LES FORMES NORMALES

Les formes normales

Objectifs

Différentes formes de normalisation de la relation universelle, le plus souvent obtenues par décomposition, afin d'obtenir un schéma de base de données qui soit (pour la 3FN) :

- sans redondance,
- sans anomalies de mise à jour,
- sans perte de données,
- sans perte de DF.

Dans ce cours : 1^{re}, 2^{e}, 3^{e} formes normales ainsi que la 3FNBC. Il existe d'autres formes normales de plus haut-niveau (intérêt plus théorique que pratique) qui ne sont pas abordées.

La première forme normale (1FN)

Définition

Une relation est en 1FN si chacun des attributs ne prend ses valeurs que dans un domaine constitué de valeurs élémentaires (*i.e.* atomiques).

La 1FN consiste à éviter les domaines composés de plusieurs valeurs (pas de données composées, pas de listes).

3

La première forme normale (1FN)

Définition

Une relation est en 1FN si chacun des attributs ne prend ses valeurs que dans un domaine constitué de valeurs élémentaires (*i.e.* atomiques).

La 1FN consiste à éviter les domaines composés de plusieurs valeurs (pas de données composées, pas de listes).

4

La deuxième forme normale (2FN)

Intérêt historique car supprime peu d'anomalies.

Définition 1

Un schéma de relation est en 2FN ssi :

- il est en 1FN,
- il n'admet pas de **dépendance de clé partielle,** c'est à dire une DF d'une partie stricte d'une clé vers des attributs non clés.

Définition 2

- 1FN
- Toutes les DF des clés vers les attributs non clés sont élémentaires.

La deuxième forme normale (2FN)

Exemple

R(<u>A, B</u>, C, D) avec B→D

6

La troisième forme normale (3FN)

Définition

Un schéma de relation est en 3FN ssi :

- il est en 2FN,
- chaque attribut non clé est pleinement et directement dépendant des clés.

ou

- tout attribut n'appartenant pas à une clé ne dépend pas d'un attribut non clé.

ou encore

- il n'admet pas de DF transitive, c'est-à-dire d'un ensemble d'attributs non inclus dans une clé vers un autre ensemble d'attributs non (sur-)clé. Toutes les DF des clés vers les attributs non clés sont élémentaires et directes.

7

La troisième forme normale (3FN)

Exemple

Avec le nouveau système d'immatriculation des véhicules, on a maintenant :
VOITURE(immatriculation, marque, type, puissance, couleur)
DF supplémentaires : type $\rightarrow$ marque et type $\rightarrow$ puissance

Est-ce que la relation VOITURE est en 3FN ?

La FNBC (ou BCNF)

Limite de la 3FN

Fournisseur(NomF, NoF, Produit, Prix) avec

NomF → NoF, NoF → NomF et NomF,Produit → Prix

Donc deux clés candidates : (NomF, Produit) et (NoF, Produit)

⇒ 3FN mais redondance dans les données entre NomF et NoF

Définition

Un schéma de relation est en FNBC (Boyce-Codd) ssi :

- aucun attribut ne dépend transitivement d'une clé
- Implique que le schéma est en 3FN

ou

- Les seules DF non triviales sont celles où une clé détermine un ou plusieurs attributs

ou

- La partie gauche des DF doit contenir une clé (être une « super-clé »)

Décomposition en 3FN et FNBC

Objectif

A partir d'une relation *r*, obtenir un schéma de base de données qui minimise le nombre de schémas de relation r_i et qui maximise leur forme normale et soit sans perte (données et dépendances).

- A partir des r_i, il doit être possible de reconstituer *r* (via des jointure).

- Il est toujours possible de décomposer en 3FN sans perte de données et sans perte de DF.

- Il est toujours possible de décomposer en FNBC sans perte de données.

10

Décomposition sans perte

Définition

La décomposition d'une relation R<U,F> en un ensemble de schémas de relations $\{R_1, ..., R_n\}$ obtenus par projection est dite sans perte si et seulement si quelque soit la réalisation r de R :

- La jointure naturelle des r_i donne exactement r (pas de perte de données) ;
- $(F_1 \cup \ldots \cup F_n)^+ = F^+$ (pas de perte de dépendances).

Théorème de Heath

Soit R(X, Y, Z) et la DF $X \rightarrow Y$ qui est vérifiée. R peut alors être décomposée en $R_1(X, Y)$ et $R_2(X, Z)$. R est égale à la jointure de ses projections sur R_1 et R_2.

Lemme de Rissanen

Soit R(X, Y, Z) et les DF $X \rightarrow Y$ et $Y \rightarrow Z$ qui sont vérifiées. R peut alors être décomposée en $R_1(X, Y)$ et $R_2(Y, Z)$.

11

Décomposition sans perte

Définition

La décomposition d'une relation R<U,F> en un ensemble de schémas de relations $\{R_1, ..., R_n\}$ obtenus par projection est dite sans perte si et seulement si quelque soit la réalisation r de R :

- La jointure naturelle des r_i donne exactement r (pas de perte de données) ;
- $(F_1 \cup \ldots \cup F_n)^+ = F^+$ (pas de perte de dépendances).

Théorème

soit un schéma de base de données S = (R’, R’’) et F l’ensemble de DF associé. S est sans perte par rapport à F ssi :

- $(R' \cap R'') \rightarrow (R' - R'') \in F^+$ ou
- $(R' \cap R'') \rightarrow (R' - R'') \in F^+$.

Exemple

Soit R(A, B, C) et $F=\{A \rightarrow B\}$ la décomposition R’ = {A, B} et R’ = {A, C} est sans perte.

12

Décomposition en FNBC

Principe

Décomposition successive (mais sans garantie de conservation des DF)

Soit une relation R à décomposer et F son ensemble de DF.

Principe

1) Si R n'est pas en FNBC, soit $X \rightarrow A \in F^+$ non triviale avec X qui n'est pas super-clé (i.e. ne contient pas une clé).

2) On décompose R en R' = R − A et R'' = (X,A). Cette décomposition est sans perte car : $R' \cap R'' = X$ et $R'' - R' = A$ et $X \rightarrow A \in F^+$.

3) On va au point 1. en considérant cette fois

— R' avec les dépendances $\pi_{R'}(F)$ et

— R'' avec les dépendances $\pi_{R''}(F)$

13

Décomposition en 3FNBC

Exemple : Soit R(C, P, H, S, E, N) avec
C : Cours, P : Professeur, H : Heure, S : Salle, E : Etudiant, N : Niveau
F={C $\rightarrow$ P, CE $\rightarrow$ N, HS $\rightarrow$ C, HE $\rightarrow$ S, HP $\rightarrow$ S}
Une seule clé : (HE)

14

Décomposition en 3FN

Définition

A partir de l'ensemble des DF fourni et en utilisant leurs propriétés, on peut directement obtenir une décomposition sans perte.

Principe

15

Conclusion sur la décomposition

• Si une relation est en FNBC, alors elle ne contient pas de redondances détectables par les DF. Donc, essayer de décomposer en BCFN d'abord.
• S'il n'est pas possible de faire la décomposition sans perte de DF, alors se contenter d'un schéma en 3FN.

• La décomposition peut être remise en cause lorsqu'on considère les coûts d'exécution des requêtes
⇒ Décomposer R en R1 et R2 implique la nécessité d'une jointure pour avoir les informations initialement dans *r* instance de R.
Ce qu'on gagne en espace de stockage, on peut le perdre au niveau de l'exécution des requêtes.

• D'où la 3ème phase de conception appelé « Dénormalisation ».

16